火龙果•整理
uml.org.cn

DTCC 2014中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2014
大数据技术探索和价值发现

大型企业级应用环境SQL优化探秘

个人介绍

- 杨廷琨(yangtingkun)
  - Oracle ACE Director
  - ITPUB数据库管理区版主
  - ACOUG核心会员
  - 参与编写《Oracle数据库性能优化》、《Oracle DBA手记》和《Oracle DBA手记3》
  - 十四年的一线DBA经验
  - 个人BLOG中积累了近3000篇原创技术文章
  - 云和恩墨技术总监

优化的思路

```
BEGIN
  LOOP
    V := F_FOUND_BOTTLENECK;
    P_OPTIMIZE(V);
    IF (F_PERFORMANCE) THEN
      EXIT;
    END IF;
  END LOOP;
END;
/
```

一条SQL语句引发的血案

Begin Snap:	20.2
End Snap:	20.3
Elapsed:	
DB Time:	

Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	191,405.24	88.44
DB CPU	14,450.68	6.68
PL/SQL execution elapsed time	255.33	0.12
sequence load elapsed time	65.57	0.03
parse time elapsed	51.94	0.02
connection management call elapsed time	19.32	0.01
hard parse elapsed time	1.45	0.00
hard parse (sharing criteria) elapsed time	1.17	0.00
hard parse (bind mismatch) elapsed time	0.36	0.00
PL/SQL compilation elapsed time	0.01	0.00
repeated bind elapsed time	0.00	0.00
DB time	216,419.36	
background elapsed time	2,068.03	
background cpu time	328.61	

Event	Wait Class
latch: cache buffers	Concurrency
CPU time	
log file sync	Commit
latch: ges resource	Other
gc buffer busy	Cluster

一条SQL语句引发的血案

SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- Total DB Time (s): 216,419
- Captured SQL account for 169.4% of Total

Elapsed Time (s)	CPU Time (s)	Executions	Elap per Exec (s)	% Total DB Time	SQL Id	SQL Module	SQL Text
161,566	13,861	38,711	4.17	74.65	88492nrstj3xd	JDBC Thin Client	BEGIN UP_CARD_PRE(:1, :2); END...
157,211	13,770	38,709	4.06	72.64	6zqqgm5k6nyt6	JDBC Thin Client	SELECT UNICARD_NO FROM TF_R_UN...
26,904	325	33,046	0.81	12.43	gup334bprcn0d	JDBC Thin Client	BEGIN UP_BANKCHARGE_CARDOPER(:...
2,982	20	33,045	0.09	1.38	fj1t7vkjn43fg	JDBC Thin Client	INSERT INTO TL_R_UNICARD_DEDUC...
2,833	17	39,169	0.07	1.31	ah95ufyrbvgpu	JDBC Thin Client	SELECT CP.ORDER_ID, CP.PREOCC_...

SQL ordered by Gets

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- Total Buffer Gets: 968,484,612
- Captured SQL account for 99.6% of Total

Buffer Gets	Executions	Gets per Exec	%Total	CPU Time (s)	Elapsed Time (s)	SQL Id	SQL Module	SQL Text
953,869,453	38,711	24,640.79	98.49	13860.52	161566.41	88492nrstj3xd	JDBC Thin Client	BEGIN UP_CARD_PRE(:1, :2); END...
951,358,528	38,709	24,577.19	98.23	13770.42	157211.08	6zqqgm5k6nyt6	JDBC Thin Client	SELECT UNICARD_NO FROM TF_R_UN...
11,436,108	33,046	346.07	1.18	324.87	26904.14	gup334bprcn0d	JDBC Thin Client	BEGIN UP_BANKCHARGE_CARDOPER(:...
3,206,309	33,053	97.01	0.33	40.85	358.92	7b4zf4mrn7p2n	JDBC Thin Client	SELECT COUNT(DEDUCT_LOGID) FRO...

一条SQL语句引发的血案

Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	652,229,833	5,305.79	52.75
CPU Time (ms)	44,931,910	365.51	66.74
Executions	122,928		
Buffer Gets	3,033,188,922	24,674.52	80.90
Disk Reads	1,763,706	14.35	4.69
Parse Calls	1,460	0.01	0.13
Rows	153,846	1.25	
User I/O Wait Time (ms)	5,543,256		
Cluster Wait Time (ms)	59,115,405		
Application Wait Time (ms)	2,854		
Concurrency Wait Time (ms)	218,479,258		
Invalidations	0		
Version Count	17		
Sharable Mem(KB)	444		

新上线SQL？
执行计划改变？

一条SQL语句引发的血案

```
SELECT UNICARD_NO
FROM TF_R_UNICARD
WHERE PRESENT_TAG = '0'
AND LIMIT_DATE + 0 > SYSDATE + 90
AND UNICARD_STATE||NULL = '0'
AND UNICARD_VALCODE||NULL = :B3
AND ROWNUM <= :B2
AND RESERVED1 = :B1
AND (RESERVED2 <> '99' OR RESERVED2 IS NULL)
FOR UPDATE
```

FOR UPDATE锁表？
LOCAL索引访问效率低？
ROWNUM绑定变量的值改变？
COUNT STOPKEY没有生效？

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT				476 (100)			
1	FOR UPDATE							
2	COUNT STOPKEY							
3	PARTITION HASH ALL		1	39	476 (1)	00:00:06	1	8
4	TABLE ACCESS BY LOCAL INDEX ROWID	TF_R_UNICARD	1	39	476 (1)	00:00:06	1	8
5	INDEX RANGE SCAN	IDX_TF_R_UNICARD_4	765		283 (1)	00:00:04	1	8

一条SQL语句引发的血案

```
SELECT UNICARD_NO
FROM TF_R_UNICARD
WHERE PRESENT_TAG = '0'
AND LIMIT_DATE + 0 > SYSDATE + 90
AND UNICARD_STATE||NULL = '0'
AND UNICARD_VALCODE||NULL = :B3
AND ROWNUM <= :B2
AND RESERVED1 = :B1
AND (RESERVED2 <> '99' OR RESERVED2 IS NULL)
FOR UPDATE

SQL> select table_name, index_name, column_name, column_position
from dba_ind_columns where table_name = 'TF_R_UNICARD';

TABLE_NAME     INDEX_NAME           COLUMN_NAME       COLUMN_POSITION
-------------  -------------------  ----------------  ---------------
TF_R_UNICARD   PK_TF_R_UNICARD      UNICARD_NO                      1
TF_R_UNICARD   IDX_TF_R_UNICARD_4   RESERVED1                       1
TF_R_UNICARD   IDX_TF_R_UNICARD_4   UNICARD_BATCHNO                 2
TF_R_UNICARD   IDX_TF_R_UNICARD_4   UNICARD_VALCODE                 3
```

一条SQL语句引发的血案

```
SQL> select column_name, NUM_DISTINCT, num_nulls, LAST_ANALYZED, HISTOGRAM from
    dba_tab_columns where table_name = 'TF_R_UNICARD';

COLUMN_NAME     NUM_DISTINCT  NUM_NULLS LAST_ANALYZED HISTOGRAM
--------------- ------------ ---------- ------------- ---------------------
UNICARD_NO         208636559          0 01-4月 -14    HEIGHT BALANCED
UNICARD_BATCHNO          548          0 01-4月 -14    HEIGHT BALANCED
UNICARD_TYPE               1          0 01-4月 -14    NONE
UNICARD_VALCODE            9          0 01-4月 -14    FREQUENCY
UNICARD_STATE              5          0 01-4月 -14    FREQUENCY
LIMIT_DATE                 6          0 01-4月 -14    FREQUENCY
BALANCE                   10          0 01-4月 -14    NONE
PRESENT_TAG                1          0 01-4月 -14    FREQUENCY
RESERVED1               1386          0 01-4月 -14    HEIGHT BALANCED
RESERVED2                  1   78524235 01-4月 -14    FREQUENCY
```

一条SQL语句引发的血案

```
SQL> SELECT CHILD_NUMBER, NAME, POSITION, DATATYPE_STRING, VALUE_STRING FROM
    V$SQL_BIND_CAPTURE WHERE SQL_ID = '6zqqgm5k6nyt6' AND CHILD_NUMBER = 0;

CHILD_NUMBER NAME  POSITION DATATYPE_STRING VALUE_STRING
------------ ----- -------- --------------- ------------
           0 :B3          1 CHAR(32)        04
           0 :B2          2 NUMBER          1
           0 :B1          3 VARCHAR2(32)    0422

SQL> SELECT UNICARD_NO
  2  FROM UCR_CARD_01.TF_R_UNICARD
  3  WHERE PRESENT_TAG = '0'
  4  AND LIMIT_DATE + 0 > SYSDATE + 90
  5  AND UNICARD_STATE||NULL = '0'
  6  AND UNICARD_VALCODE||NULL = '04'
  7  AND ROWNUM <= 1
  8  AND RESERVED1 = '0422'
  9  AND (RESERVED2 <> '99' OR RESERVED2 IS NULL) ;

UNICARD_NO
----------------------------------------
```

一条SQL语句引发的血案

```
Elapsed: 00:00:02.71

Statistics
----------------------------------------------------------
          0  recursive calls
          0  db block gets
      14079  consistent gets
          0  physical reads
          0  redo size
        530  bytes sent via SQL*Net to client
        492  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed
```

一条SQL语句引发的血案

```
Execution Plan
----------------------------------------------------------------------------------------
|Id|Operation                                |Name                |Rows |Cost|Pstart|Pstop|
----------------------------------------------------------------------------------------
| 0|SELECT STATEMENT                         |                    |    1| 947|      |     |
|*1| COUNT STOPKEY                           |                    |     |    |      |     |
| 2|  PARTITION HASH ALL                     |                    |    1| 947|    1 |    8|
|*3|   TABLE ACCESS BY LOCAL INDEX ROWID|TF_R_UNICARD        |    1| 947|    1 |    8|
|*4|    INDEX RANGE SCAN                     |IDX_TF_R_UNICARD_4| 1484| 609|    1 |    8|
----------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
   1 - filter(ROWNUM<=1)
   3 - filter(("RESERVED2" IS NULL OR "RESERVED2"<>'99') AND
   "UNICARD_STATE"||NULL='0' AND
              INTERNAL_FUNCTION("LIMIT_DATE")+0>SYSDATE@!+90 AND "PRESENT_TAG"='0')
   4 - access("RESERVED1"='0422')
       filter("UNICARD_VALCODE"||NULL='04')
```

一条SQL语句引发的血案

```
SQL> SELECT UNICARD_NO
  2  FROM UCR_CARD_01.TF_R_UNICARD PARTITION (P1) A
  3  WHERE PRESENT_TAG = '0'
  4  AND LIMIT_DATE + 0 > SYSDATE + 90
  5  AND UNICARD_STATE||NULL = '0'
  6  AND UNICARD_VALCODE||NULL = '04'
  7  AND ROWNUM <= 1
  8  AND RESERVED1 = '0422'
  9  AND (RESERVED2 <> '99' OR RESERVED2 IS NULL) ;

no rows selected
```

```
Statistics
----------------------------------------------------------
          1  recursive calls
          0  db block gets
       2662  consistent gets
          0  physical reads
          0  redo size
        327  bytes sent via SQL*Net to client
        481  bytes received via SQL*Net from client
          1  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          0  rows processed
```

```
Execution Plan
----------------------------------------------------------------------------------------
| Id  | Operation                          |Name              |Rows |Pstart|Pstop|
----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |                  |    1|      |     |
|*  1 |  COUNT STOPKEY                     |                  |     |      |     |
|   2 |   PARTITION HASH SINGLE            |                  |    1|    1 |    1|
|*  3 |    TABLE ACCESS BY LOCAL INDEX ROWID|TF_R_UNICARD      |    1|    1 |    1|
|*  4 |     INDEX RANGE SCAN               |IDX_TF_R_UNICARD_4|   11|    1 |    1|
----------------------------------------------------------------------------------------
```

一条SQL语句引发的血案

```
SQL> SELECT /*+ GATHER_PLAN_STATISTICS */ UNICARD_NO
  2  FROM UCR_CARD_01.TF_R_UNICARD
  3  WHERE PRESENT_TAG = '0'
  4  AND LIMIT_DATE + 0 > SYSDATE + 90
  5  AND UNICARD_STATE||NULL = '0'
  6  AND UNICARD_VALCODE||NULL = '04'
  7  AND ROWNUM <= 1
  8  AND RESERVED1 = '0422'
  9  AND (RESERVED2 <> '99' OR RESERVED2 IS NULL) ;

UNICARD_NO
----------------------------------------
XXXXXXXXXXXXXX
```

一条SQL语句引发的血案

```
SQL> SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY_CURSOR(NULL, NULL, 'IOSTATS'));

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------
SQL_ID  gcm5adh9hr10a, child number 0
-------------------------------------
SELECT /*+ GATHER_PLAN_STATISTICS */ UNICARD_NO FROM UCR_CARD_01.TF_R_UNICARD WHERE PRESENT_TAG =
    '0' AND LIMIT_DATE + 0 > SYSDATE + 90 AND UNICARD_STATE||NULL = '0' AND UNICARD_VALCODE||NULL =
    '04' AND ROWNUM <= 1 AND RESERVED1 = '0422' AND (RESERVED2 <> '99' OR RESERVED2 IS NULL)

Plan hash value: 490980256

-------------------------------------------------------------------------------------------------
| Id  | Operation                          |Name              |Starts|E-Rows|A-Rows|Buffers |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |                  |     1|      |     1|  14079 |
|*  1 |  COUNT STOPKEY                     |                  |     1|      |     1|  14079 |
|   2 |   PARTITION HASH ALL               |                  |     1|     1|     1|  14079 |
|*  3 |    TABLE ACCESS BY LOCAL INDEX ROWID|TF_R_UNICARD      |     5|     1|     1|  14079 |
|*  4 |     INDEX RANGE SCAN               |IDX_TF_R_UNICARD_4|     5|  1484| 19366|   1023 |
-------------------------------------------------------------------------------------------------
```

一条SQL语句引发的血案

```
SQL> SELECT UNICARD_NO
  2  FROM UCR_CARD_01.TF_R_UNICARD PARTITION(P1)
  3  W SQL> SELECT UNICARD_NO
  4  A   2  FROM UCR_CARD_01.TF_R_UNICARD PARTITION(P2)
  5  A   3  W SQL> SELECT UNICARD_NO
  6  A   4  A   2  FROM UCR_CARD_01.TF_R_UNICARD PARTITION(P3)
  7  A   5  A   3  W SQL> SELECT UNICARD_NO
  8  A   6  A   4  A   2  FROM UCR_CARD_01.TF_R_UNICARD PARTITION(P4)
  9  A   7  A   5  A   3  W SQL> SELECT UNICARD_NO
         8  A   6  A   4  A   2  FROM UCR_CARD_01.TF_R_UNICARD PARTITION(P5)
no row   9  A   7  A   5  A   3  WHERE PRESENT_TAG = '0'
                8  A   6  A   4  AND LIMIT_DATE + 0 > SYSDATE + 90
      no row    9  A   7  A   5  AND UNICARD_STATE||NULL = '0'
                     8  A   6  AND UNICARD_VALCODE||NULL = '04'
            no row   9  A   7  AND ROWNUM <= 1
                         8  AND RESERVED1 = '0422'
                 no row  9  AND (RESERVED2 <> '99' OR RESERVED2 IS NULL) ;

                 UNICARD_NO
                 ----------------------------------------
                 XXXXXXXXXXXXXX
```

一条SQL语句引发的血案

```
SELECT UNICARD_NO
FROM TF_R_UNICARD
WHERE PRESENT_TAG = '0'
AND LIMIT_DATE + 0 > SYSDATE + 90
AND UNICARD_STATE||NULL = '0'
AND UNICARD_VALCODE||NULL = :B3
AND ROWNUM <= :B2
AND RESERVED1 = :B1
AND (RESERVED2 <> '99' OR RESERVED2 IS NULL)
FOR UPDATE

SQL> ALTER SESSION FORCE PARALLEL DDL;

Session altered.

SQL> CREATE INDEX IND_UNICARD_RES_VALCODE_DATE ON TF_R_UNICARD
  2  (RESERVED1, UNICARD_VALCODE||NULL, UNICARD_STATE||NULL, LIMIT_DATE + 0) PARALLEL 8 ONLINE;

Index created.

SQL> ALTER INDEX IND_UNICARD_RES_VALCODE_DATE NOPARALLEL;
```

一条SQL语句引发的血案

Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	652,229,833	5,305.79	52.75
CPU Time (ms)	44,931,910	365.51	66.74
Executions	122,928		
Buffer Gets	3,033,188,922	24,674.52	80.90
Disk Reads	1,763,706	14.35	4.69
Parse Calls	1,460	0.01	0.13
Rows	153,846	1.25	
User I/O Wait Time (ms)	5,543,256		
Cluster Wait Time (ms)	59,115,405		
Application Wait Time (ms)	2,854		
Concurrency Wait Time (ms)	218,479,258		
Invalidations	0		
Version Count	17		
Sharable Mem(KB)	444		

Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	1,662,915	245.63	0.56
CPU Time (ms)	280,230	41.39	1.79
Executions	6,770		
Buffer Gets	20,870,067	3,082.73	2.45
Disk Reads	10,246	1.51	1.41
Parse Calls	64	0.01	0.08
Rows	6,769	1.00	
User I/O Wait Time (ms)	48,971		
Cluster Wait Time (ms)	62,598		
Application Wait Time (ms)	4		
Concurrency Wait Time (ms)	304,772		
Invalidations	4		
Version Count	6		
Sharable Mem(KB)	54		

一条SQL语句引发的血案

- 针对业务特点的表结构模型
- 合理的分区方案
- 根据访问方式设计索引

一条SQL语句引发的血案

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
latch: cache buffers chains	90,884	67,851	747	31.4	Concurrency
CPU time		14,451		6.7	
log file sync	202,004	13,134	65	6.1	Commit
latch: ges resource hash list	26,429	8,347	316	3.9	Other
gc buffer busy	23,770	7,143	300	3.3	Cluster

```
SQL> select 31.4 + 6.7 + 6.1 + 3.9 + 3.3 from dual;

31.4+6.7+6.1+3.9+3.3
--------------------
                  51.4
```

N条SQL语句引发的血案

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	35113	28-Feb-14 09:30:12	1180	27.6
End Snap:	35115	28-Feb-14 10:30:41	1923	51.9
Elapsed:		60.47 (mins)		
DB Time:		26,393.90 (mins)		

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
CPU time		124,175		7.8	
log file sync	780,264	30,408	39	1.9	Commit
gc buffer busy	520,629	24,024	46	1.5	Cluster
gc current block 2-way	1,666,951	20,109	12	1.3	Cluster
gc cr block 2-way	1,611,054	18,402	11	1.2	Cluster

时间都去哪了？

N条SQL语句引发的血案

Elapsed Time (s)	CPU Time (s)	Executions	Elap per Exec (s)	% Total DB Time	SQL Id	SQL Module	SQL Text
719,535	3,898	22,659	31.75	45.44	5uu4w5h2s7vrw	JDBC Thin Client	SELECT /*+ use_nl(a, b)*/a.PRO...
117,899	760	63,604	1.85	7.44	1npv7hb39m5uu	JDBC Thin Client	select distinct(B.CODE), B.NAM...
81,128	8,915	8,981	9.03	5.12	djs1vvfbx384g	JDBC Thin Client	SELECT PRODUCT_TYPE_CODE, PROD...
74,794	370	24,576	3.04	4.72	32vm6qm1fa54d	JDBC Thin Client	select t1.*, f_csm_getelementL...
58,671	6,657	11,529	5.09	3.70	6gpmtr8rqg5sd	JDBC Thin Client	SELECT A.PRODUCT_ID, PRODUCT_N...
43,301	150	10,679	4.05	2.73	10gkc95bsnr0b	JDBC Thin Client	SELECT distinct TRADE_TYPE_COD...
27,319	87	7,126	3.83	1.73	bbq12yrgj4djz	JDBC Thin Client	SELECT to_char(a.TRADE_ID) TRA...
21,821	6,581	7,791	2.80	1.38	0ghfzggnu4kku	JDBC Thin Client	select * from (select row_.*, ...
19,810	46	1,678	11.81	1.25	bvmavt81hjux4	JDBC Thin Client	SELECT distinct a.product_type...
19,384	56	4,810	4.03	1.22	5p3d8gtuwy9yn	JDBC Thin Client	SELECT PARA_CODE1 , PARA_CODE2...

CPU Time (s)	Elapsed Time (s)	Executions	CPU per Exec (s)	% Total	% Total DB Time	SQL Id	SQL Module	SQL Text
8,915	81,128	8,981	0.99	7.18	5.12	djs1vvfbx384g	JDBC Thin Client	SELECT PRODUCT_TYPE_CODE, PROD...
8,349	8,738	33,183	0.25	6.72	0.55	g23td847a4y3g	JDBC Thin Client	SELECT s.object_id STAFFID, s....
7,389	12,281	3,545	2.08	5.95	0.78	b0sv14d2zmpxh	JDBC Thin Client	BEGIN P_SDR_TODAY_TRADE_003(:1...
6,657	58,671	11,529	0.58	5.36	3.70	6gpmtr8rqg5sd	JDBC Thin Client	SELECT A.PRODUCT_ID, PRODUCT_N...
6,581	21,821	7,791	0.84	5.30	1.38	0ghfzggnu4kku	JDBC Thin Client	select * from (select row_.*, ...
6,432	6,845	25,560	0.25	5.18	0.43	1079w18dav83v	JDBC Thin Client	SELECT s.object_id STAFFID, s....
4,943	8,983	2,296	2.15	3.98	0.57	5m019d9pcnggs	JDBC Thin Client	select * from (select row_.*, ...
4,178	6,105	2,422	1.72	3.36	0.39	gs8bu90vkmu73	JDBC Thin Client	INSERT INTO TM_R_TRADE_003 (VA...
3,898	719,535	22,659	0.17	3.14	45.44	5uu4w5h2s7vrw	JDBC Thin Client	SELECT /*+ use_nl(a, b)*/a.PRO...

N条SQL语句引发的血案

Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	8,740,969,835	21,757.18	51.95
CPU Time (ms)	69,690,400	173.47	2.67
Executions	401,751		
Buffer Gets	242,148,091	602.73	0.21
Disk Reads	53,849	0.13	0.02
Parse Calls	357,264	0.89	0.26
Rows	6,004,330	14.95	
User I/O Wait Time (ms)	507,306		
Cluster Wait Time (ms)	482,208		
Application Wait Time (ms)	0		
Concurrency Wait Time (ms)	120,168		
Invalidations	0		
Version Count	212		
Sharable Mem(KB)	4,105		

N条SQL语句引发的血案

	Snap Id	Snap Time
Begin Snap:	35113	28-Feb-14 09:30:12
End Snap:	35115	28-Feb-14 10:30:41
Elapsed:		60.47 (mins)
DB Time:		26,393.90 (mins)

Event	Waits	Time(s)	Avg Wait(
CPU time		124,175	
log file sync	780,264	30,408	
gc buffer busy	520,629	24,024	
gc current block 2-way	1,666,951	20,109	
gc cr block 2-way	1,611,054	18,402	

Statistic	Total
AVG_BUSY_TIME	322,162
AVG_IDLE_TIME	38,246
AVG_IOWAIT_TIME	4,829
AVG_SYS_TIME	28,915
AVG_USER_TIME	293,042
BUSY_TIME	12,895,347
IDLE_TIME	1,537,483
IOWAIT_TIME	200,636
SYS_TIME	1,164,710
USER_TIME	11,730,637
LOAD	2
OS_CPU_WAIT_TIME	3.6E+12
RSRC_MGR_CPU_WAIT_TIME	0
VM_IN_BYTES	58,535,920
VM_OUT_BYTES	7,979,000
PHYSICAL_MEMORY_BYTES	2.1E+11
NUM_CPUS	40
NUM_CPU_SOCKETS	40

```
SQL> select 40*3600 from dual;

        40*3600
---------------
          144000
```

N条SQL语句引发的血案

stattime	cpu_total	mem_util	disk_util	swap_space	net_util	DISK_FS_IO_RATE	NET_PACKET_RATE
09:00:02,	81.7,	86.1,	8.1,	48.0,	4.92,	12.7,	62162.5
09:05:02,	87.3,	86.3,	8.9,	48.0,	5.57,	12.7,	74992.7
09:10:02,	90.6,	86.4,	16.2,	48.0,	5.06,	12.6,	74005.4
09:15:02,	94.2,	86.5,	19.7,	48.0,	5.11,	12.6,	72529.2
09:20:02,	97.0,	86.6,	15.1,	48.0,	5.08,	12.6,	80785.3
09:25:02,	97.0,	86.7,	13.3,	48.0,	5.94,	12.6,	92448.6
09:30:02,	95.9,	86.9,	13.0,	49.0,	4.92,	13.5,	71388.7
09:35:02,	96.5,	86.9,	14.3,	49.0,	5.13,	13.5,	75115.2
09:40:02,	94.9,	86.8,	12.9,	49.0,	5.80,	13.4,	80465.1
09:45:02,	96.9,	86.8,	13.7,	49.0,	5.89,	13.4,	79514.0
09:50:02,	99.3,	86.9,	9.9,	49.0,	5.33,	13.3,	75420.0
09:55:02,	99.1,	87.1,	14.5,	49.0,	5.56,	13.3,	79440.6
10:00:02,	98.1,	87.0,	17.8,	49.0,	6.04,	13.3,	82996.8
10:05:02,	100.0,	88.6,	21.8,	54.0,	6.86,	13.9,	97903.1
10:16:15,	71.4,	90.8,	79.8,	54.0,	3.05,	16.7,	51089.0
10:20:47,	74.7,	90.4,	80.6,	54.0,	3.99,	17.8,	67753.6
10:25:00,	75.0,	90.4,	81.4,	53.0,	4.05,	18.8,	69295.5
10:29:14,	73.5,	90.2,	81.8,	53.0,	3.46,	19.7,	54465.0

N条SQL语句引发的血案

Elasped Time = DB CPU + ON WAIT + RUN QUEUE

火龙果•整理
uml.org.cn

N条SQL语句引发的血案

- CPU扩容
- 增加RAC节点
- 迁移新服务器
- 整体SQL全面优化
- 部分非关键性业务限流

总结

- 突然爆发型
  - 易于定位
  - 解决迅速
  - 上线前SQL审核
- 温水煮青蛙型
  - 问题定位困难
  - 解决方案涉及结构和硬件的改变
  - 建立关键SQL运行基线
  - 记录系统硬件阈值，便于及时升级扩容
  - 深入的性能分析和评估

火龙果•整理
uml.org.cn

Q&A

THANKS

IT168.com ChinaUnix ITPUB